Implementation and validation of an Oxygen Exchange model coupled with multiple physics

life^x project

Teresa Babini

Manfred Nesti

Supervisors: Dr. P. C. Africa, Dr. I. Fumagalli, Dr. F. Regazzoni

Advanced Programming for Scientific Computing
Politecnico di Milano

11/02/2022

Myocardium model

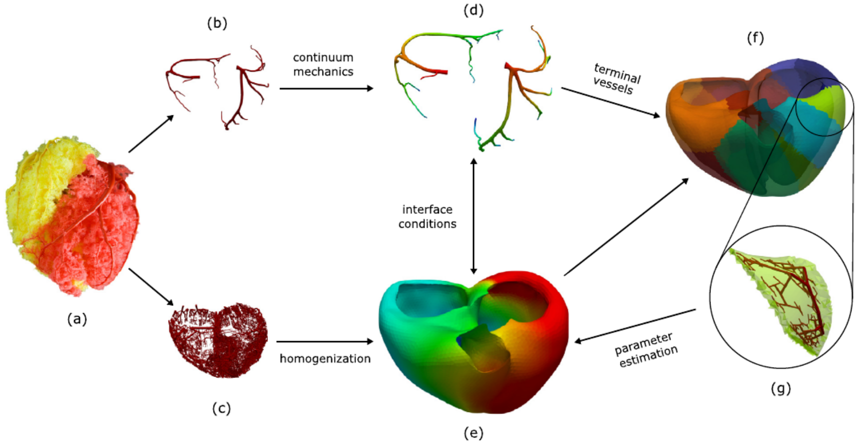


Figure: (a) Cast of Coronary Arteries, source Wikipedia; (b) epicardial vessels; (c) intramural vessels; (d) 3D blood flow dynamics model inside coronary domain (pressure depicted); (e) porous media flow model inside myocardial domain (pressure depicted); (f) myocardium partitioned into different perfusion regions; (g) example of generated intramural vascular network inside a perfusion region. Source: [3].

Oxygen Exchange model (OE)

$$\frac{\partial}{\partial t}\,\mathrm{PO_2}^i + \nabla\cdot(\mathrm{PO_2}^i\,\psi_i^{-1}\hat{\mathbf{u}}_{\mathbf{i}}) = n\alpha\,[\mathrm{Hb}*]\,k_-\left(\mathrm{SO_2}^i - (1-\mathrm{SO_2}^i)\left(\frac{\mathrm{PO_2}^i}{\mathrm{PO_2}^{50}}\right)^n\right)$$
$$+\,\psi_i^{-1}\hat{\phi}_{i-1,i}\,\mathrm{PO_2}^{i-1} - \psi_i^{-1}\hat{\phi}_{i,i+1}\,\mathrm{PO_2}^i$$
$$-\,\psi_i^{-1}\tilde{P}\left(\mathrm{PO_2}^i - \mathrm{PO_2}^m\right)\delta_{i3}$$

$$\frac{\partial}{\partial t}\,\mathrm{SO_2}^i + \nabla\cdot(\mathrm{SO_2}^i\,\psi_i^{-1}\hat{\mathbf{u}}_{\mathbf{i}}) = -k_-\left(\mathrm{SO_2}^i - (1-\mathrm{SO_2}^i)\left(\frac{\mathrm{PO_2}^i}{\mathrm{PO_2}^{50}}\right)^n\right)$$
$$+\,\psi_i^{-1}\hat{\phi}_{i-1,i}\,\mathrm{SO_2}^{i-1} - \psi_i^{-1}\hat{\phi}_{i,i+1}\,\mathrm{SO_2}^i$$

$$\frac{\partial}{\partial t}\,\mathrm{PO_2}^m = \psi_m^{-1}\tilde{P}(\mathrm{PO_2}^3 - \mathrm{PO_2}^m) - \tilde{\xi}_0\left(1 + \frac{\mathrm{PO_2}^{m,50}}{\mathrm{PO_2}^m}\right)^{-1}$$

$$\Lambda_{\mathrm{O_2}}^{\mathrm{del}} = \frac{1}{T|\Omega|}\int_0^T\int_\Omega \tilde{P}\alpha^{-1}(\mathrm{PO_2}^3 - \mathrm{PO_2}^m)$$

$$\Lambda_{\mathrm{O_2}}^{\mathrm{cons}} = \frac{1}{T|\Omega|}\int_0^T\int_\Omega \psi_m\tilde{\xi}_0\alpha^{-1}\left(1 + \frac{\mathrm{PO_2}^{m,50}}{\mathrm{PO_2}^m}\right)^{-1}$$

OE model - Reduced model

We can define the total oxygen concentration as $[O_2^*]^i := n\,[Hb^*]\,SO_2{}^i + \alpha^{-1}\,PO_2{}^i$ getting the reduced model

$$\frac{\partial}{\partial t}[O_2^*]^i + \nabla\cdot\left([O_2^*]^i\,\psi_i^{-1}\hat{\mathbf{u}}_i\right) = \psi_i^{-1}\hat{\phi}_{i-1,i}\,[O_2^*]^{i-1} - \psi_i^{-1}\hat{\phi}_{i,i+1}\,[O_2^*]^i$$

$$- \psi_i^{-1}\tilde{P}\alpha^{-1}\left(\left(g^{-1}([O_2^*]^i\right) - PO_2{}^m\right)\delta_{i3}$$

$$\frac{\partial}{\partial t}PO_2{}^m = \psi_m^{-1}\tilde{P}\left(g^{-1}([O_2^*]^3) - PO_2{}^m\right) - \tilde{\xi}_0\left(1 + \frac{PO_2{}^{50}}{PO_2{}^m}\right)^{-1}$$

$$\Lambda_{O_2}^{\text{del}} = \frac{1}{T|\Omega|}\int_0^T\int_\Omega \tilde{P}\alpha^{-1}\left(g^{-1}([O_2^*]^3) - PO_2{}^m\right)$$

$$\Lambda_{O_2}^{\text{cons}} = \frac{1}{T|\Omega|}\int_0^T\int_\Omega \psi_m\tilde{\xi}_0\alpha^{-1}\left(1 + \frac{PO_2{}^{m,50}}{PO_2{}^m}\right)^{-1}$$

Uncoupled problem

$$\hat{\mathbf{u}}_3 = \mathbf{0}$$

$$\hat{\phi}_{2,3} = \Phi \sin\left(\pi \frac{t \bmod T_{HB}}{T_F}\right)^4 1\left((t \bmod T_{HB}) < T_F\right)$$

$$\hat{\phi}_{3,4} = \hat{\phi}_{3,\text{veins}}(t) = \hat{\phi}_{2,3}(t)$$

where $\Phi = 0.05\ \text{s}^{-1}$ is the flux magnitude, $T_{HB} = 0.8$ s is the period of a heart beat and $T_F = 0.6$ s is the duration of the simulated flux.

Coupled problem

$$\hat{\mathbf{u}}_3 = -K_3 \nabla p_3$$

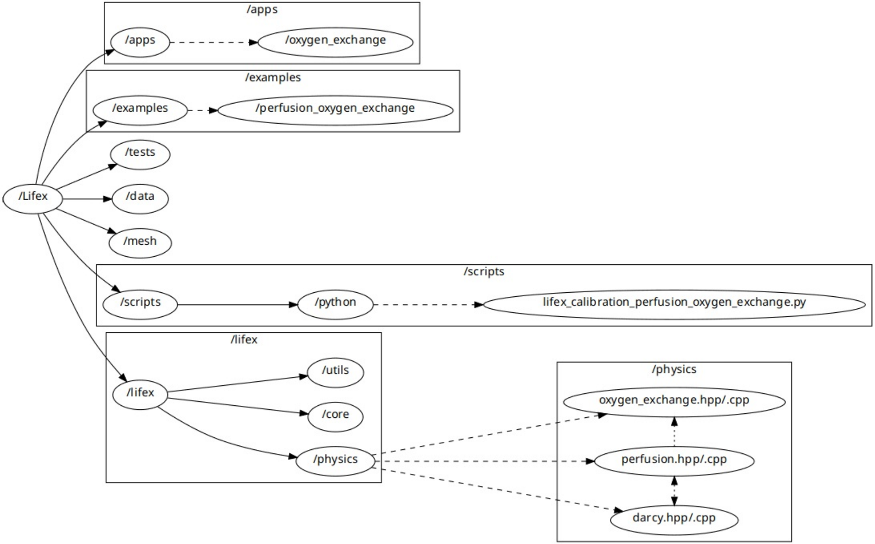
$$\hat{\phi}_{2,3} = \beta_{23}(p_3 - p_2)$$

$$\hat{\phi}_{3,4} = \hat{\phi}_{3,\text{veins}} = \gamma(p_3 - p_{\text{veins}})$$

where K_3 is the permeability tensor of the 3$^{\text{rd}}$ compartment, $\beta_{2,3}$ and γ are the inter-compartment pressure-coupling coefficients and p_{veins} is veins pressure.

$$
\begin{aligned}
\int_\Omega \frac{[\mathrm{O}_2^*]^{3,k+1}}{\Delta t} v_h d\omega &- \int_\Omega \mu \nabla [\mathrm{O}_2^*]^{3,k+1} \nabla v_h d\omega \\
&- \int_\Omega [\mathrm{O}_2^*]^{3,k+1} \psi_3^{-1} \hat{\mathbf{u}}_3 \cdot \nabla v_h d\omega + \int_\Omega \psi_3^{-1} \hat{\phi}_{3,4} [\mathrm{O}_2^*]^{3,k+1} v_h d\omega \\
&= \int_\Omega \frac{[\mathrm{O}_2^*]^{3,k}}{\Delta t} v_h d\omega + \int_\Omega \psi_3^{-1} \hat{\phi}_{2,3} [\mathrm{O}_2^*]^{2,k+1} v_h d\omega \\
&- \int_\Omega \psi_3^{-1} \tilde{P} \alpha^{-1} \left(g^{-1} \left([\mathrm{O}_2^*]^{3,k} \right) - \mathrm{PO_2}^{m,k} \right) v_h d\omega \quad \forall\ v_h \ \in X_h \\
\frac{\mathrm{PO_2}^{m,k+1}}{\Delta t} &= \frac{\mathrm{PO_2}^{m,k}}{\Delta t} + \psi_m^{-1} \tilde{P} \left(g^{-1} \left([\mathrm{O}_2^*]^{3,k} \right) - \mathrm{PO_2}^{m,k} \right) \\
&- \tilde{\xi}_0 \left(1 + \frac{\mathrm{PO_2}^{50}}{\mathrm{PO_2}^{m,k}} \right)^{-1}
\end{aligned}
$$

life^x library




```
module load gcc-glibc/11 dealii vtk
```

OxygenExchange class - Constructor

```
namespace lifex
{
  OxygenExchange::OxygenExchange(const std::string &subsection,
                                 const bool &standalone_)
    : CoreModel(subsection)
    , standalone(standalone_)
    , triangulation(std::make_shared<utils::MeshHandler>(
        prm_subsection_path,
        mpi_comm,
        std::initializer_list<utils::MeshHandler::GeometryType↩
          >(
          {utils::MeshHandler::GeometryType::File,
          utils::MeshHandler::GeometryType::Hypercube})))
    , fe_scalar(1)
    , linear_solver(prm_subsection_path + " / Linear solver",
                    {"CG", "GMRES", "BiCGStab"},
                    "GMRES")
    , block_preconditioner(prm_subsection_path +
                             " / Block preconditioner",
                           system_matrix)
    , csv_writer(mpi_rank == 0)
  {}
```

OxygenExchange class - Insights

- We override the following base methods

```
declare_parameters()
parse_parameters()
run()
```

- To build some non-default set of parameters, we used `config/nondefault.json` file inside our app and example
- The `run` method calls in sequence different methods, depending on the spatial dimension
- To contain the 3D model solutions we used `LinAlg::MPI::BlockVector` data structures

```
LinAlg::MPI::BlockVector
  c_O2_star; ///< [O2*] solution vector with ghost entries.
LinAlg::MPI::BlockVector
  PO2_muscle; ///< PO2_muscle solution vector with ghost ←
     entries.
```

OxygenExchange class - Mathematical issues

- To handle the nonlinearity of the problem we used a look-up table based on a `std::map<double, double> g_inv_table` as it follows

```
/// Inverse of g.
double
g_inv(const double &c_O2_star) const
{
    auto   p  = g_inv_table.lower_bound(c_O2_star);
    double y2 = p->first;
    double x2 = p->second;
    // undefined behavior for prev if p = table.begin()
    double y1 = std::prev(p, 1)->first;
    double x1 = std::prev(p, 1)->second;
    return ((x2 - x1) / (y2 - y1)) * (c_O2_star - y1) + x1;
}
```

- We declared the class `Perfusion` as `friend` of `OxygenExchange` for coupling reason

```
class OxygenExchange : public CoreModel
{ public: // ...
friend class Perfusion;
```

DarcyLinear	OxygenExchange
$p_3,\ p_2,\ p_{veins},\ K_3,\ \beta_{2,3}$	$\phi_{2,3}, \phi_{3,veins},\ \hat{\mathbf{u}}_3 = -K_3 \nabla p_3$

```
void
Perfusion::update_oxygen_exchange()
{
oxygen_exchange->set_flux_darcy(darcy->get_gamma(),
                                darcy->get_p_veins(),
                                darcy->get_beta_23(),
                                darcy->get_p_owned());
oxygen_exchange->set_p_3_owned(darcy->get_p_owned());
}
```

```
grad_p_3 =
    std::make_unique<QuadratureFEMGradient>(
                                p_3,
                                dof_handler_scalar,
                                *quadrature_formula);
```

Perfusion class - BoundaryScalar

```
/// Class in charge of computing outlet pressure for Network.
class BoundaryScalar : public ScalarBC
{
// ...
virtual double
value(const double & /*t*/)
{
  if (network_flow)
    return darcy->compute_average_subdomain_pressure(
      region, 0); // Always on first compartment
  else
    return 0;
};
```

Perfusion class - New example

```
/// Scalar source of the Darcy problem.
class ExampleScalarSource : public ScalarSource
{ // ...
virtual double
value(const Point<dim> &p, const unsigned int component = 0) ↩
    const override
{
  double out = 0; // Output placeholder
  if (component == 0)
    { ); // Coupling with network flow
      out += values.at(material_id)
      // Other sources
      if (inlet_source_type == "Uniform")
        out += val;
      else if (inlet_source_type == "Sphere")
        { // ... }
      else if (inlet_source_type == "Sinusoidal")
        { // ... }
    }
  if (component == veins_compartment)
    out += venous_return;
  return out;
}
```

Spherical inlet for Darcy model.

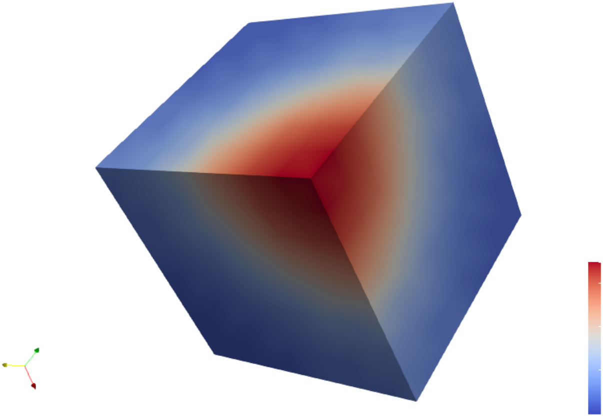
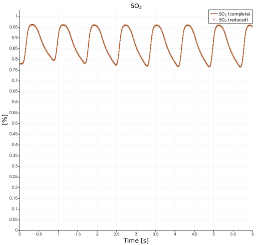
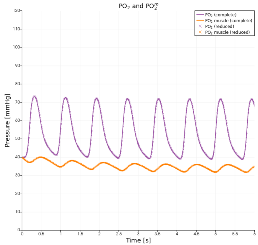


Figure: Spherical inlet for Darcy model

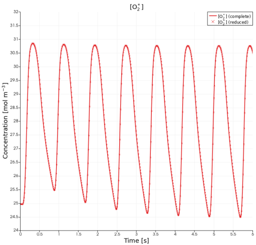
Comparison between complete and reduced 0D models



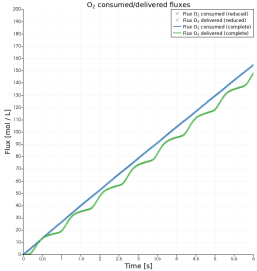
(a) SO_2



(b) PO_2 and $PO_2{}^m$



(c) $[O_2^*]$



(d) O_2 fluxes

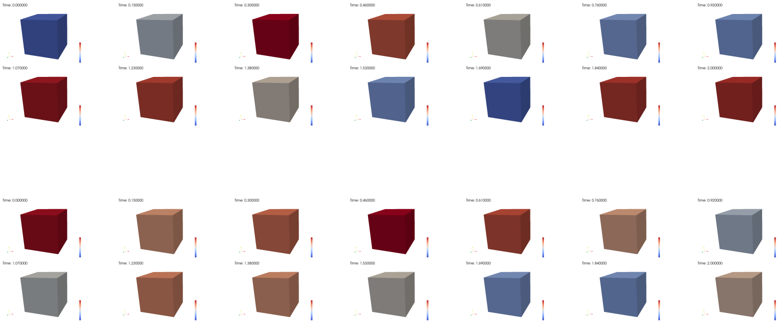
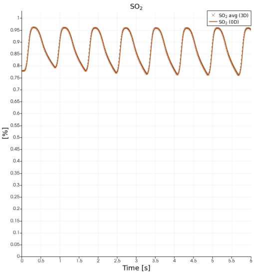
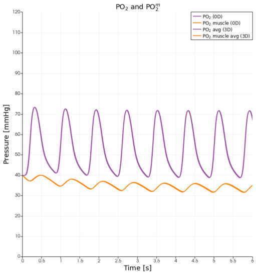


Figure: Results of 3D uncoupled model ($[O_2^*]$ and PO_2) with $t \in [0, 2]$ s, one frame every ~ 0.15 s.

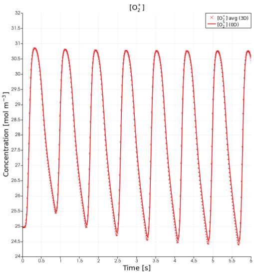
Comparison between reduced 3D and 0D models



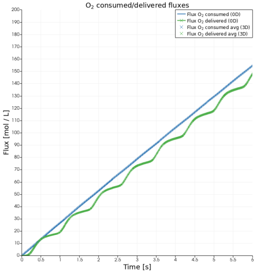
(a) SO_2



(b) PO_2 and $PO_2{}^m$



(c) $[O_2^*]$



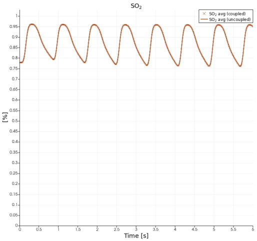
(d) O_2 fluxes

Calibration of Darcy source

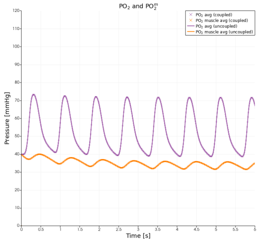
```python
import numpy as np; import pandas as pd; from prettytable import PrettyTable
# Parameters
beta_12 = 2.6e-6; beta_23 = 3.62e-6; gamma = 1e-4
p_veins = 22.5 * 133; sphere_r = 5e-3; cube_l = 1e-2
sphere_vol = 1/8 * 4/3 * np.pi * sphere_r**3; cube_vol = cube_l**3
# Compute inlet source value
delta_p_12 = 24.66 * 133 # (from mmHg to Pa)
phi_01 = delta_p_12 * beta_12
value = phi_01 / (sphere_vol / cube_vol)
# Results
df = pd.read_csv('./results/perfusion.csv')
p_perf = np.array([df["p0_avg"][1], df["p1_avg"][1], df["p2_avg"][1]])
p = np.array([p_veins + gamma**(-1) * phi_01,
    p_veins + gamma**(-1) * phi_01 + beta_23**(-1) * phi_01,
    p_veins + gamma**(-1) * phi_01 + beta_23**(-1) * phi_01 + beta_12**(-1) * phi_01])
delta_p = (p_perf - p) / p_perf # ...
```

	Compartment 1	Compartment 2	Compartment 3
Avg pressure [Pa]	3077.77	5433.42	8713.2
Perfusion pressure [Pa]	8981.75	5548.0	3081.78
Delta [%]	0.66	0.02	-1.83

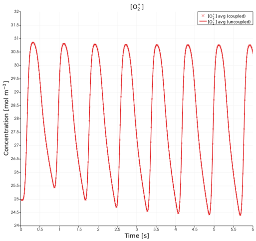
Comparison between coupled and 3D (uncoupled) models



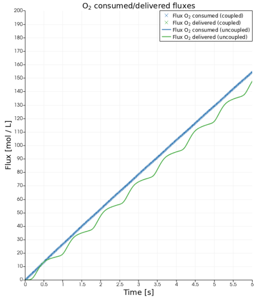
(a) SO_2



(b) PO_2 and $PO_2{}^m$

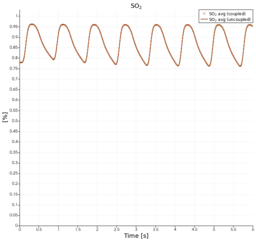


(c) $[O_2^*]$

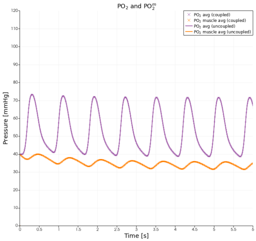


(d) O_2 fluxes

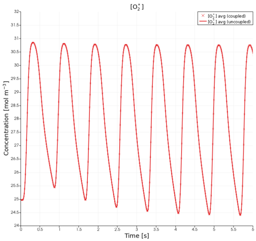
Validation of coupled 3D model - Smaller volume



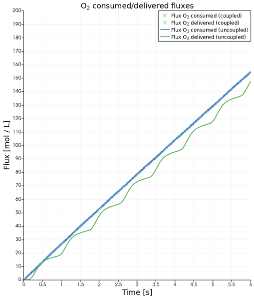
(a) SO_2



(b) PO_2 and $PO_2{}^m$

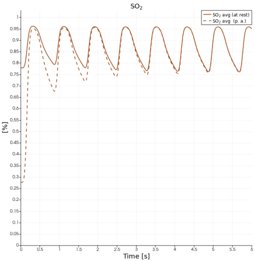


(c) $[O_2^*]$

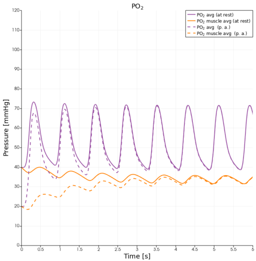


(d) O_2 fluxes

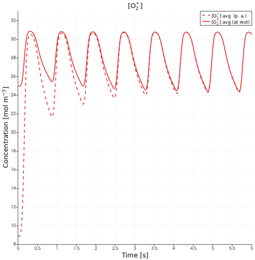
Effect of physical activity



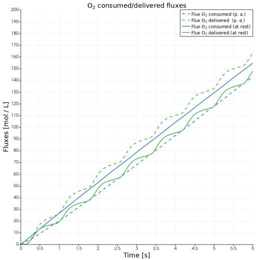
(a) SO_2



(b) PO_2 and $PO_2{}^m$

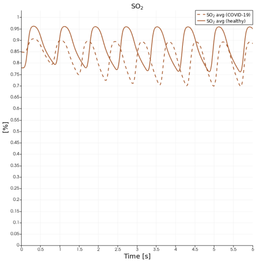


(c) $[O_2^*]$

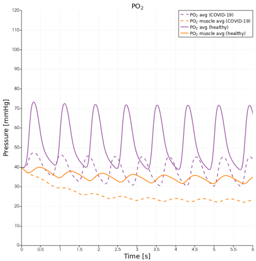


(d) O_2 fluxes

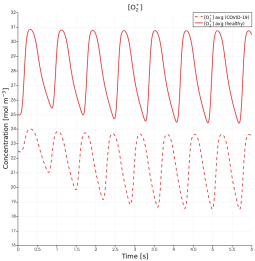
Effect of COVID-19 disease



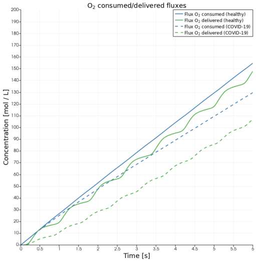
(a) SO_2



(b) PO_2 and $PO_2{}^m$

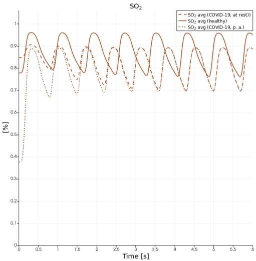


(c) $[O_2^*]$

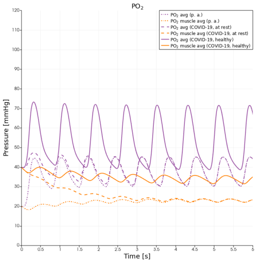


(d) O_2 fluxes

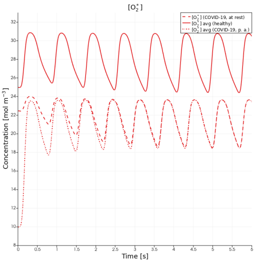
Effect of physical activity and COVID-19 disease



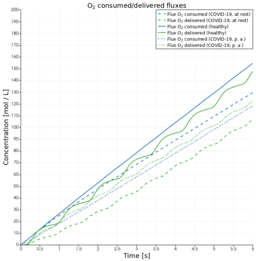
(a) SO_2



(b) PO_2 and ${PO_2}^m$

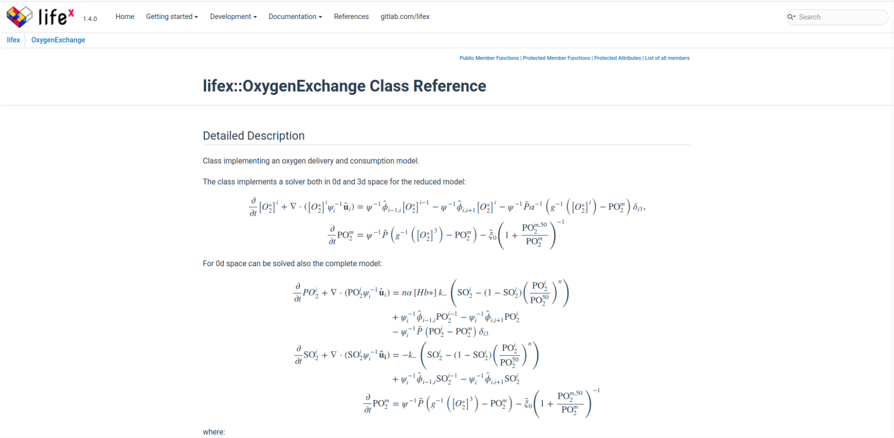


(c) $[O_2^*]$



(d) O_2 fluxes

https://gitlab.com/lifex/lifex/-/merge_requests/291



life^x 1.4.0 Home Getting started Development Documentation References gitlab.com/lifex Search

lifex OxygenExchange

Public Member Functions | Protected Member Functions | Protected Attributes | List of all members

lifex::OxygenExchange Class Reference

Detailed Description

Class implementing an oxygen delivery and consumption model.

The class implements a solver both in 0d and 3d space for the reduced model:

$$\frac{\partial}{\partial t}[O_2^*]^i + \nabla\cdot\left([O_2^*]^i\psi_i^{-1}\hat{\mathbf{u}}_i\right) = \psi^{-1}\hat{\phi}_{i-1,i}[O_2^*]^{i-1} - \psi^{-1}\hat{\phi}_{i,i+1}[O_2^*]^i - \psi^{-1}\tilde{P}\alpha^{-1}\left(g^{-1}\left([O_2^*]^i\right) - \mathrm{PO}_2^m\right)\delta_{i3},$$
$$\frac{\partial}{\partial t}\mathrm{PO}_2^m = \psi^{-1}\tilde{P}\left(g^{-1}\left([O_2^*]^3\right) - \mathrm{PO}_2^m\right) - \tilde{\xi}_0\left(1 + \frac{\mathrm{PO}_2^{m,50}}{\mathrm{PO}_2^m}\right)^{-1}$$

For 0d space can be solved also the complete model:

$$\begin{aligned}
\frac{\partial}{\partial t}PO_2^i + \nabla\cdot\left(\mathrm{PO}_2^i\psi_i^{-1}\hat{\mathbf{u}}_i\right) &= n\alpha\,[Hb*]\,k_-\left(\mathrm{SO}_2^i - (1-\mathrm{SO}_2^i)\left(\frac{\mathrm{PO}_2^i}{\mathrm{PO}_2^{50}}\right)^n\right)\\
&\quad + \psi_i^{-1}\hat{\phi}_{i-1,i}\mathrm{PO}_2^{i-1} - \psi_i^{-1}\hat{\phi}_{i,i+1}\mathrm{PO}_2^i\\
&\quad - \psi_i^{-1}\tilde{P}\left(\mathrm{PO}_2^i - \mathrm{PO}_2^m\right)\delta_{i3}\\
\frac{\partial}{\partial t}\mathrm{SO}_2^i + \nabla\cdot\left(\mathrm{SO}_2^i\psi_i^{-1}\hat{\mathbf{u}}_\mathrm{i}\right) &= -k_-\left(\mathrm{SO}_2^i - (1-\mathrm{SO}_2^i)\left(\frac{\mathrm{PO}_2^i}{\mathrm{PO}_2^{50}}\right)^n\right)\\
&\quad + \psi_i^{-1}\hat{\phi}_{i-1,i}\mathrm{SO}_2^{i-1} - \psi_i^{-1}\hat{\phi}_{i,i+1}\mathrm{SO}_2^i\\
\frac{\partial}{\partial t}\mathrm{PO}_2^m &= \psi^{-1}\tilde{P}\left(g^{-1}\left([O_2^*]^3\right) - \mathrm{PO}_2^m\right) - \tilde{\xi}_0\left(1 + \frac{\mathrm{PO}_2^{m,50}}{\mathrm{PO}_2^m}\right)^{-1}
\end{aligned}$$

where:

Figure: OxygenExchange class documentation in life^x website

Essential bibliography

- F. Regazzoni – *Oxygen Exchange model*, internal report, Politecnico di Milano (2020).
- P. C. Africa, R. Piersanti, M. Fedele, L. Dede', A. Quarteroni – *life^x - heart module: a high-performance simulator for the cardiac function*, https://arxiv.org/abs/2201.03303, Politecnico di Milano (2022).
- S. Di Gregorio, M. Fedele, G. Pontone, A. F. Corno, P. Zunino, C. Vergara, and A. Quarteroni – *A computational model applied to myocardial perfusion in the human heart: From large coronaries to microvasculature*, Journal of Computational Physics 424 (2021) 109836.
- life^x documentation – https://lifex.gitlab.io/lifex/index.html
- deal.ii deocumentation – https://dealii.org/